

**APPLICATION FOR
UNITED STATES PATENT
IN THE NAME OF**

**PETER A. HAWKINS
COLIN N.B. COOK**

Assigned to

INTEL CORPORATION

for

**STAR INTELLIGENT PLATFORM
MANAGEMENT BUS TOPOLOGY**

Prepared by:

Pillsbury Winthrop LLP
725 South Figueroa Street, Suite 2800
Los Angeles, California 90017-5406
Tel: (213) 488-7100
Fax: (213) 629-1033

ATTORNEY DOCKET NO. 81674-249758

Client Docket No.: P-13079

Express Mail No.: EL 860 912 823 US

2007-10-29 10:44:00

STAR INTELLIGENT PLATFORM MANAGEMENT BUS TOPOLOGY

BACKGROUND

5 **1. Field of the Invention**

Embodiments described herein are directed to a star Intelligent Platform Management Bus (“IPMB”) topology. Specifically, the star IPMB topology uses independent IPMBs between a central Baseboard Management Controller (“BMC”) and multiple satellite management controllers (“SMCs”). An SMC is any management controller that is not the central BMC. As
10 such, an SMC may or may not include BMC functionality.

2. Related Art

In response to central processing unit (“CPU”) performance problems and the need for total server availability, multiple server vendors developed the Intelligent Platform Management Interface (“IPMI”) standard. IPMI is an open standard hardware manageability interface
15 specification that defines how unique devices can communicate with a CPU in a standard fashion. With IPMI, a CPU makes requests and receives information from an IPMI event log through a Baseboard Management Controller (“BMC”). The devices communicate in a standard manner with the IPMI event log, whereby the CPU only inquires about changes in the event log since the previous inquiry. With IPMI, use of the CPU is minimized, thereby allowing overall
20 system performance improvements. IPMI provides a cost-effective and efficient way for a server’s CPU to communicate with the devices it needs to monitor.

The IPMI standard includes the following elements: the IPMI, the Intelligent Platform Management Bus (“IPMB”), the Intelligent Chassis Management Bus (“ICMB”), standard message and data formats, satellite management controllers (“SMCs”), and the BMC. The IPMI
25 is the specification for the management controller command sets, including command sets for

sensors, event logs, and sensor data record access, as well as the specification for the data formats, including sensor data records, event log entries, and Field Replaceable Unit ("FRU") inventory information. IPMI is also the name used for the overall standardization effort.

The IPMB is an inter-integrated circuit ("I²C")-based, multi-master bus used for intra-chassis communication with SMCs. The ICMB is the RS-485 (TIA/EIA Recommended Standard 485A, published March 1, 1998) based inter-chassis management bus, based on IPMB. It is used for common chassis and emergency management functions, including power and reset control, chassis status, events, and FRU inventory. RS-485 is a standard for multipoint communications.

The BMC is used to monitor baseboard temperatures and voltages and to manage the system event log and non-volatile storage for sensor data records. It provides a system software interface to the IPMB. The BMC further manages the interface between the system management software and the platform management hardware, provides recovery control, and serves as a gateway between system management software and the IPMB and the ICMB.

In a typical IPMI based system, a single IPMB provides a communication connection between all of the IPMI controllers in the system. The BMC is one such controller. The other controllers are SMCs. SMCs are management controllers that are distributed within the system, away from a central BMC.

A dual IPMB architecture using two buses to help eliminate single points of failure in a system has been developed. There remains a need, however, for a topology that uses independent IPMBs between the BMC and the various SMCs so as to extend platform management and provide advantages over the standard bus implementation of typical IPMI based systems.

BRIEF DESCRIPTION OF THE DRAWINGS

A detailed description of embodiments of the invention will be made with reference to the accompanying drawings, wherein like numerals designate corresponding parts in the several figures.

5 FIG. 1 is a block diagram of the main components of a computer system supporting an intelligent platform management interface, according to an embodiment of the present invention.

FIG. 2 is an illustration of the architecture of a star intelligent platform management bus, according to an embodiment of the present invention.

DETAILED DESCRIPTION

10 The following paragraphs describe a star intelligent platform management bus ("IPMB") 110 topology. IPMB 110 is an inter-integrated circuit ("I²C")-based bus that provides a standardized interconnection between different modules within a chassis. Intelligent devices that use IPMB 110 are typically controllers that perform platform management functions. Platform management refers to the monitoring and control functions that are built into platform hardware and are primarily used for monitoring the health of system hardware. This generally includes monitoring elements such as system temperatures, voltages, fans, power supplies, bus errors, and system physical security. It also includes automatic and manually driven recovery capabilities such as local or remote system resets and power on/off operations. It further includes the logging of abnormal or out-of-range conditions for later examination and alerting where the platform issues the alert without aid of run-time software. Moreover, it includes inventory information that can help identify a failed hardware unit.

20 Figure 1 shows the main components of an Intelligent Platform Management Interface ("IPMI") 100 (current version 1.5, revision 1.0, February 21, 2001, published by Intel

Corporation, Hewlett-Packard Company, NEC Corporation, and Dell Computer Corporation).

IPMI **100** is a hardware level interface specification that is management software neutral, providing monitoring and control functions that can be exposed through standard management software interfaces such as, for example, Desktop Management Interface (“DMI”), Windows

5 Management Instrumentation (“WMI”), Common Information Model (“CIM”), and Simple Network Management Protocol (“SNMP”). As a hardware level interface, it sits at the bottom of a typical management software stack. IPMI **100** is best used in conjunction with system management software running under the operating system. This provides an enhanced level of manageability by providing in-band access to the IPMI **100** management information and
 10 integrating IPMI **100** with the additional management functions provided by management applications and the operating system.

At the heart of the IPMI **100** architecture is a microcontroller known as the Baseboard Management Controller (“BMC”) **120**. The BMC **120** provides the intelligence behind intelligent platform management. The BMC **120** manages the interface between system
 15 management software and the platform management hardware, provides autonomous monitoring, event logging, and recovery control and serves as the gateway between system management software and the IPMB **110** and an Intelligent Chassis Management Bus (“ICMB”) **140**.

The BMC **120** is connected to a system bus **190** on the computer chassis motherboard
 20 **200** through a system interface **160**. The motherboard **200** is then connected to a network controller and a network connector such as a Local Area Network (“LAN”) or Wide Area Network (WAN”) connector, which is then coupled to a LAN or a WAN.

The BMC **120** may include or be connected to a non-volatile storage unit **170**. This non-

volatile storage unit **170** may include a system event log, a sensor data record repository, and a baseboard Field Replaceable Unit (“FRU”) information database. The BMC **120** may further include or be connected to sensors and control circuitry **180** for monitoring voltages, temperatures, power, reset control, fans, etc. The BMC **120** may also be connected to private management buses that are coupled to memory and processor modules.

IPMI **100** supports the extension of platform management by connecting additional satellite management controllers (“SMCs”) **130** to the system using the IPMB **110**. IPMI’s **100** support for multiple SMCs **130** shows that the architecture is scalable. The SMCs **130** reside on a chassis module **150**. SMCs **130** receive command messages from the BMC **120**. The SMC **130** accepts the command message from the BMC **120**, gathers the information from the appropriate device such as a sensor, packages the information in the appropriate transmission format, and transmits the response message with the information over the IPMB **110** to the BMC **120**. The SMC **130** may also send event messages to the BMC **120** if a sensor detects an event. Events may include out-of-range values, crossed thresholds, etc. If the SMC **130** interface is located in another computer chassis module **150**, messages are sent over the ICMB **140**.

ICMB **140** is the RS-485 (TIA/EIA Recommended Standard 485A, published March 1, 1998) based inter-chassis management bus, based on IPMB **110**. It is used for common chassis and emergency management functions, including power and reset control, chassis status, events, and FRU inventory. RS-485 is the standard for multipoint communications. Coupled to the ICMB **140** is an ICMB **140** bridge that includes RS-485 transceivers as well as a microcontroller.

The IPMB **110** (current version 1.0, revision 1.0, November 15, 1999, published by Intel Corporation, Hewlett-Packard Company, NEC Corporation, and Dell Computer Corporation) is the standardized bus and protocol for extending management control, monitoring, and event

delivery within the chassis. The IPMB 110 is used for communication to and between the various controllers such as the BMC 120 and the SMC 130.

The IPMB 110 architecture and protocol addresses several goals. The IPMB 110 supports a distributed platform management architecture. Sensors and controllers may be located on the managed modules and their information consolidated via the IPMB 110. This yields a more flexible design than one in which all sensors must be directly routed to a central point of management.

The IPMB 110 further supports asynchronous event notification and critical event logging. The IPMB 110 implements a multi-master protocol that allows intelligent controllers to arbitrate the bus for the purpose of sending an event message to an event receiver node. This provides a mechanism whereby a controller can raise an asynchronous event.

The IPMB 110 provides an extensible platform management infrastructure. New management information sources can be readily added to the IPMB 110 without impacting other controllers on the bus. The IPMB 110 implements a multi-master operation to support the distributed management architecture, asynchronous event notification, and platform extensibility. The mechanism supports direct communication between any two intelligent devices on the bus.

The IPMB 110 is designed to allow non-intelligent I²C devices to co-reside on the IPMB 110. Such devices, including I/O ports for example, may be incorporated as part of the platform management system. Such devices can be accessed directly or can be managed as devices that are owned by an intelligent controller.

The IPMB 110 is separate from the system's processor and memory buses. As such, it remains available even if a failure prevents the system from running. It is possible for the IPMB 110 to be augmented by system management add-in cards, such as autonomous management

cards that connect to the management bus and allow management data to be delivered to a remote console via a telephone line or a LAN connection.

The IPMB 110 provides an inexpensive, low-pin count, communication media for platform management information. Miscellaneous system cabling functions, such as the routing of fault signals between modules, can be replaced by using the IPMB 110. A dedicated wire is typically used only to communicate a single piece of management information, whereas the IPMB 110 carries whole streams of data. Moreover, the ICMB 140 protocol provides a route to interchassis management. This is accomplished by store-and-forward type devices referred to as bridge nodes. Bridge nodes isolate the internal management bus address spaces in order to eradicate any concern about address conflicts between the internal nodes in one chassis and those in another.

Figure 2 illustrates a star IPMB 110 topology. BMC 120 is shown connected to SMCs 130a-e. Instead of a single IPMB 110 providing a communication connection between the BMC 120 and the SMCs 130a-e, independent IPMBs 110a-e are implemented. This star IPMB 110 topology offers several features over the standard bus implementation. Fault isolation is one such improvement over typical IPMI 100 based systems. That is, if SMC 130a fails in such a way that it corrupts IPMB 110a to which it is connected, a bused implementation would lose all communication among SMCs 130a-e and the BMC 120. With a star IPMB 110 topology, this type of failure would only cause communication to be lost with the failed SMC 130a.

An additional advantage of the star IPMB 110 topology is that it offers separate address domains. In a typical IPMI 100 based bused system, the BMC 120 and all SMCs 130a-e must have a unique address. With a star IPMB 110 topology, multiple controllers can potentially have the same address. This feature is especially useful in rack systems, *e.g.* Compact Peripheral

Component Interconnect (“CompactPCI”), where multiple modules may implement several BMCs 120, which are all addressed at bus address 20h.

Moreover, the star IPMB 110 topology offers multiple owner security. In a rack system such as that of CompactPCI, multiple owners may have modules in the same chassis. In a bused system, one owner or a hacker compromising one owner’s module can send IPMI 100 commands to another owner’s module, possibly causing undesirable behavior. Even with encryption and authentication in a bused system, an owner or a hacker can present a denial of service (“DOS”) attack by flooding the common IPMB 110 with messages. The star IPMB 110 topology isolates each module so that a module’s controller can only directly communicate with the central BMC 120 for the chassis.

While the above description refers to particular embodiments of the present invention, it will be understood to those of ordinary skill in the art that modifications may be made without departing from the spirit thereof. The accompanying claims are intended to cover any such modifications as would fall within the true scope and spirit of the present invention.

The presently disclosed embodiments are therefore to be considered in all respects as illustrative and not restrictive; the scope of the invention being indicated by the appended claims, rather than the foregoing description. All changes that come within the meaning and range of equivalency of the claims are therefore intended to be embraced therein.